

previous time step.

We note that this algorithm requires that a strategy with the greatest anticipated benefit can be computed efficiently. The following lemma provides this requirement when the input describes the states explicitly. Note that we cannot apply this lemma to the Web server farm problem since here the states are implicit and of exponential size, so we use the offline algorithm described in copending patent application Serial No. ~~09~~^{10/000,320} (IBM Docket YOR9-2000-0824US1).

10/000,320
8/31/07

Lemma 1. *A strategy with the largest anticipated benefit can be computed in time that is polynomial in the number of states.*

Proof. Use dynamic programming that goes interactively over the time steps and accumulate discounted benefits. As the additional benefit of the next time step depends only on the current state, it suffices to have a table of size proportional to the number of states.

Theorem 2. *The above discounting algorithm has competitive ratio at most*

$$\left(1 + \frac{1}{L}\right)^{L\sqrt{L+1}}.$$

Proof. Consider the online algorithm at time t , and denote by b_0, b_1, \dots, b_L the sequence of benefits in the strategy with the largest anticipated benefit over all strategies that the online algorithm has at time t . Define $ON_t = b_0$ and

$ON_{t+1}^{*L} = b_1/\alpha + \dots + b_L/\alpha^L$, then $ON_t + ON_{t+1}^{*L}$ is the largest anticipated benefit over all the strategies that are available to the online algorithm at time t . Since

10000149-120401